Accepted for the 2019 International Joint Conference on Neural Networks

Not All Adversarial Examples Require a Complex Defense: Identifying Over-optimized Adversarial Examples with IQR-based Logit Thresholding

Utku Ozbulak[1, 3] Arnout Van Messem[2, 3] Wesley De Neve[1, 3]
[1]*Department of Electronics and Information Systems, Ghent University, Belgium*
[2]*Department of Applied Mathematics, Computer Science and Statistics, Ghent University, Belgium*
[3]*Center for Biotech Data Science, Ghent University Global Campus, Republic of Korea*
{utku.ozbulak,arnout.vanmessem,wesley.deneve}@ugent.be

***Abstract*—Detecting adversarial examples currently stands as one of the biggest challenges in the field of deep learning. Adversarial attacks, which produce adversarial examples, increase the prediction likelihood of a target class for a particular data point. During this process, the adversarial example can be further optimized, even when it has already been wrongly classified with 100% confidence, thus making the adversarial example even more difficult to detect. For this kind of adversarial examples, which we refer to as *over-optimized* adversarial examples, we discovered that the logits of the model provide solid clues on whether the data point at hand is adversarial or genuine. In this context, we first discuss the masking effect of the softmax function for the prediction made and explain why the logits of the model are more useful in detecting over-optimized adversarial examples. To identify this type of adversarial examples in practice, we propose a non-parametric and computationally efficient method which relies on interquartile range, with this method becoming more effective as the image resolution increases. We support our observations throughout the paper with detailed experiments for different datasets (MNIST, CIFAR-10, and ImageNet) and several architectures.**

I. Introduction

Even though deep convolutional architectures outperform other models on various image-related problems such as image classification [1], object detection [2], and segmentation [3], it has been shown that these models are not foolproof. A recent development called *adversarial examples* currently counts as one of the major issues these models are facing. Adversarial examples were first discovered by [4] while searching for intriguing properties of neural networks. Although there is no clear definition of *adversarial example*, we will say that a data point is called adversarial if it is perturbed to be misclassified.

The earliest methods to generate adversarial examples mostly rely on optimizing an input for a specific target class using a gradient-based algorithm, hence forcing the data point under consideration to be classified in another category [4, 5]. To improve the effectiveness of adversarial example generation, a variety of methods based on different optimization techniques were introduced. Some examples are: finding the closest distance to the decision boundary [6], Fast-Gradient Sign (FGS) [7], Jacobian-based Saliency Maps (JSMA) [8], and the recently introduced *Carlini & Wagner attack* (CW) [9]. The latter is currently regarded as the strongest attack, given that it can incorporate defense mechanisms into the optimization procedure, thus making it possible to generate strong adversarial examples that are able to circumvent these defense mechanisms [9].

As attacks get more effective with every study, new defense mechanisms are also introduced to counter them. The intuitive approach of adversarial re-training was the first method tested to prevent this problem [7]. [10] applied adversarial re-training on ImageNet [11] and found that it counters certain attacks. Another mechanism to prevent adversarial examples was introduced by [12], showing that network distillation provides some defense against adversarial examples generated with JSMA. [13] suggested extracting a binary threshold from the output of each rectifier layer (ReLU) and utilizing these thresholds with a quantized radial basis support vector machine detector to distinguish adversarial examples from genuine images. The most impactful research on adversarial defense mechanisms has thus far been performed by [14], who showed that none of the proposed defense mechanisms so far are effective against strong attacks and that the properties of currently available defense mechanisms do not scale well to higher resolution images, e.g., from MNIST images to ImageNet images.

[14] further argued that defense mechanisms should be evaluated against strong attacks and that they need to demonstrate that white-box attacks can be prevented. In this context, an attack is said to be white-box in nature if the attacker has access to the specifications of the model (weights and

classes) and is aware of the defense mechanism. However, assuming the attacker has access to both the model and the defense mechanism presents an unfair challenge to the defender. Indeed, the knowledge about the model and the defense mechanism can then be incorporated into an attack, making it possible to generate adversarial examples that have been carefully designed to avoid the defense mechanism under consideration. In their evaluation of defense techniques, [14] put two constraints on the generation of adversarial examples, namely (1) discretization and (2) the amount of perturbation. We believe this approach may lead to results that need to be carefully interpreted because, as we will show in this paper, it is trivial to distinguish natural images from heavily optimized adversarial examples based on logit values. Therefore, we claim that not all adversarial examples necessarily require a complex defense mechanism and that, in fact, a significant number of them can already be detected by analyzing the logits of the prediction.

Contributions: In summary, our work makes the following contributions:

- We discuss the masking effect of the softmax function, hiding the magnitude of the predicted logit value. This observation lays the foundation for a novel technique for identifying adversarial examples, as discussed in more detail in the remainder of this paper.
- We show that attack mechanisms may generate what we call *over-optimized* adversarial examples. These adversarial examples, which have extremely high logit outputs, can be easily identified when the logits of the prediction are analyzed.
- We introduce a non-parametric method to calculate a logit threshold from training data. This threshold can then be used to identify over-optimized adversarial examples.
- When the resolution of an image increases, we show that the subspace in which adversarial examples can be generated also increases substantially. Furthermore, unlike some of the proposed defense techniques, our method is able to identify more and more adversarial examples as the resolution of the given image increases. We provide evidence for these observations by making use of the MNIST, CIFAR-10, and ImageNet datasets.

II. Impact of Adversarial Optimization on Logits

Methods for adversarial example generation use specific optimization techniques to increase the chance a given input is labeled with the targeted class. Most of the proposed optimization techniques are iterative in nature. Indeed, it has been shown that these techniques produce more effective adversarial examples with less perturbation, compared to single-step optimization techniques or techniques that facilitate untargeted attacks [15].

We can simplify the behavior of targeted iterative optimization techniques as follows: assume that we have an affine classifier in a two-dimensional setting, $f(\mathbf{x}) : \mathbb{R}^2 \rightarrow \mathbb{R}$, separating the input space into two subspaces $f(\mathbf{x}) > 0$ and $f(\mathbf{x}) < 0$, with the line $f(\mathbf{x}) = 0$ denoting the decision boundary. This setting is shown in Fig. 1. Under these circumstances, when the given example α_0 is iteratively optimized towards $f(x) > 0$, the distances $\Delta_{\{1,2,3\}}$ between the generated adversarial points $\alpha_{\{1,2,3\}}$ and the decision boundary are increased at each step, with the goal to increase the likelihood that the data point under consideration (that is, α_0) is classified as $f(\mathbf{x}) > 0$. To that end, it does not matter whether the attack uses the logits of the model (like CW) or not, given that the overall impact of the optimization techniques used on the logits remains the same. A more detailed explanation of adversarial optimization and its impact on the decision boundary for multi-class classifiers can be found in [6].

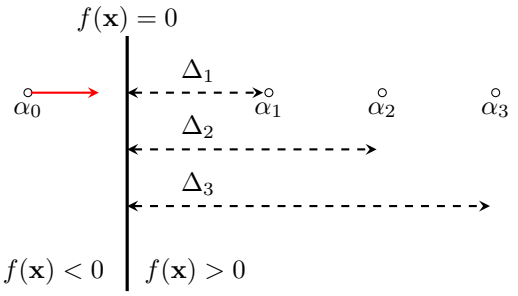


Fig. 1: An illustration of adversarial examples $\alpha_{\{1,2,3\}}$, derived from an input α_0, for an affine classifier $f(x) = 0$, and their respective distances $\Delta_{\{1,2,3\}} = d(\alpha_{\{1,2,3\}}, f)$ to the decision boundary f.

In this study, we consider two targeted attacks: (1) Basic Iterative Method [16], a fast attack that arguably generates *weak* adversarial examples, and (2) CW, a slow attack that generates *strong* adversarial examples.

A. Basic Iterative Method

Basic Iterative Method, also referred to as Iterative Fast Gradient Sign (I-FGS), finds its origin in the FGS method proposed in [7]. FGS performs a one-step update on the input along the direction of the gradient:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \alpha \ \text{sign}(\nabla_x J(g(\theta, \mathbf{x}_n)_c)\,,$$

where $\text{sign}(\nabla_x J(.))$ is the signature of the gradient of the cross-entropy error function and α is the perturbation multiplier. This method is characterized as *fast* since it does not require an iterative approach.

[16] extended FGS into I-FGS, proposing to use a lower value for α and to iteratively update the input image:

$$\mathbf{x}_{n+1} = Clip_{\mathbf{X},\epsilon}(\mathbf{x}_n - \alpha \ \text{sign}(\nabla_x J(g(\theta, \mathbf{x}_n)_c)\,.$$

In the above equation, the clipping function makes sure that the resulting adversarial example is a valid image. Furthermore, the parameter ϵ controls the maximally allowed amount of perturbation per pixel.

FGS, along with its extension I-FGS, is one of the most commonly used adversarial attacks to test adversarial defense mechanisms. However, in this study, we opted for I-FGS over FGS because I-FGS is shown to produce stronger adversarial examples [16].

(a) Original Image
Prediction: Arctic Fox
Confidence: 0.99
Logit_1: ~ 20
Logit_2: ~ 5

(b) Adv. Image
Prediction: Radio
Confidence: 1
Logit_1: $\sim 1e2$
Logit_2: ~ 10

(c) Adv. Image
Prediction: Radio
Confidence: 1
Logit_1: $\sim 5e2$
Logit_2: ~ 12

(d) Adv. Image
Prediction: Radio
Confidence: 1
Logit_1: $\sim 4e3$
Logit_2: ~ 40

Fig. 2: (a) Original image, predicted as *arctic fox* with 0.99 confidence. (b)-(c)-(d) Over-optimized adversarial examples which are predicted with the same confidence but with vastly different logit values by ResNet-50. Logit_1 and Logit_2 represent the logits of the most likely and second most likely predictions, respectively.

B. Carlini & Wagner Attack

Even though FGS and I-FGS can generate adversarial examples with a high success rate, the main focus of these methods is not to produce *strong* adversarial examples. In [9], Carlini and Wagner proposed a heavily optimized attack which produces strong adversarial examples that can bypass defense mechanisms easily. In this study, we use the L_2 version of CW, which was also used by Carlini and Wagner to test the robustness of defense techniques [14]. The L_2 version of CW is defined as follows:

$$\text{miminize} \quad ||\mathbf{x} - (\mathbf{x} + \delta)||_2^2 + \alpha\, \ell(\mathbf{x} + \delta)\,,$$

where this attack attempts to find a small perturbation δ that is sufficient to change the prediction made by the model when it is added to the input, while keeping the L_2 distance between the original image $\mathbf{x}$ and the perturbed image $\mathbf{x}' = \mathbf{x} + \delta$ minimal.

In their original work [9], they discussed multiple loss functions for this kind of attack, and in this study, we use the loss function they preferred in their later work [14] to evaluate multiple defense mechanisms. This loss function ℓ is constructed as follows:

$$\ell(\mathbf{x}') = \max\left\{\max\{g(\theta, \mathbf{x}')_i : i \neq c\} - g(\theta, \mathbf{x}')_c, -K\right\}\,, \tag{1}$$

comparing the logit prediction of the target class with the logit prediction of the next-most-likely class. This method can also be tuned using K to adjust the strength of the adversarial example produced. Specifically, as K increases, the confidence of the prediction for the adversarial example also increases. They refer to this kind of adversarial examples as *high-confidence adversarial examples*.

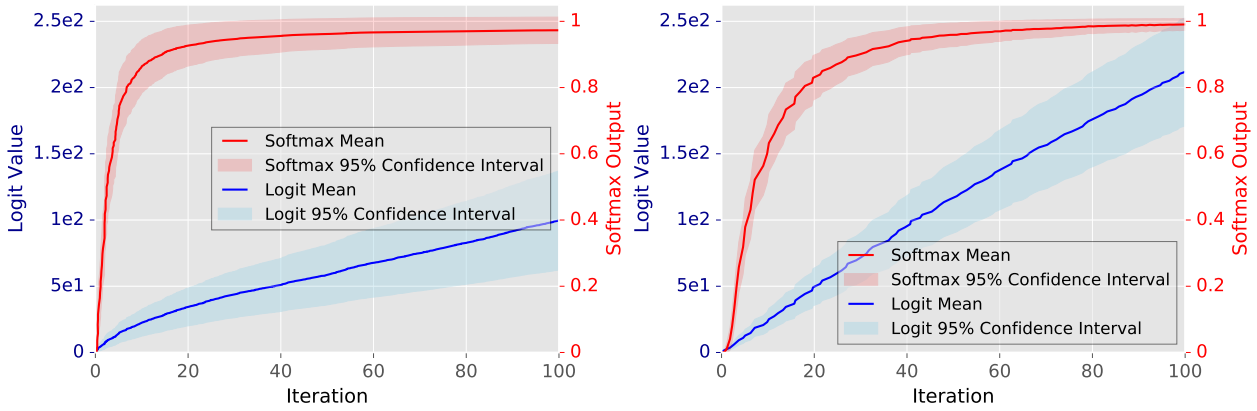
III. Masking Effect of the Softmax Function

When the prediction of a neural network is analyzed, the output is usually represented in terms of probabilities. Such probability is usually referred to as the confidence of the prediction made. To convert logit values into probabilities, a normalized exponential function called the softmax function $P(\mathbf{u})_k = \dfrac{e^{\mathbf{u}_k}}{\sum_{m=1}^{M} e^{\mathbf{u}_m}}$ is used, where $\mathbf{u}$ is an input vector such that $\mathbf{u} = (u_1, \ldots, u_M)^T \in \mathbb{R}^M$ and k is the selected index of the vector $\mathbf{u}$ [17, 18, 19]. In particular, the softmax function uses the exponential function to squeeze the input values between zero and one in such a way that the output values add up to one. This property makes the softmax function helpful in more easily interpreting the predictions of a neural network, instead of having to rely on the logits, which are more difficult to interpret.

The usage of the softmax function for convolutional neural networks dates back to 1998 [20], soon thereafter becoming a common tool to convert logits into probabilitistic values. However, when investigating adversarial examples, the softmax function, when used in settings with limited decimal precision, has a major drawback for correctly interpreting the predictions of a neural network: it lacks sensitivity to positive changes in the magnitude of the largest logit. We detailed this discovery and its impact on adversarial examples in a previous study [21].

A. Experimental Results on the Masking Effect of Softmax

To show a practical outcome of the aforementioned limitation of the softmax function for neural networks, we present Fig. 4, showing a mock-up two-class classification problem that visualizes the masking effect of the softmax function. A neural network $\hat{\mathbf{y}} = g(\theta, \mathbf{x})$ with a single hidden layer and ReLU activations is applied to this problem. The approximate decision boundary is indicated using a dashed line. For those points lying on the orange line ($x_2 = 0$), Fig. 4 displays both the predicted logit value $\max(g(\theta, \mathbf{x}))$ and the predicted outcome probability $\max(P(g(\theta, \mathbf{x})))$. It can be seen from Fig. 4 that the logit values increase as the distance between the point under consideration and the classification boundary



(a) I-FGS (b) CW

Fig. 3: Highest logit value and corresponding softmax output as a function of the number of iterations when generating adversarial examples with (a) I-FGS and (b) CW. Adversarial examples are tested on ResNet-50 in a white-box setting. Best viewed in color.

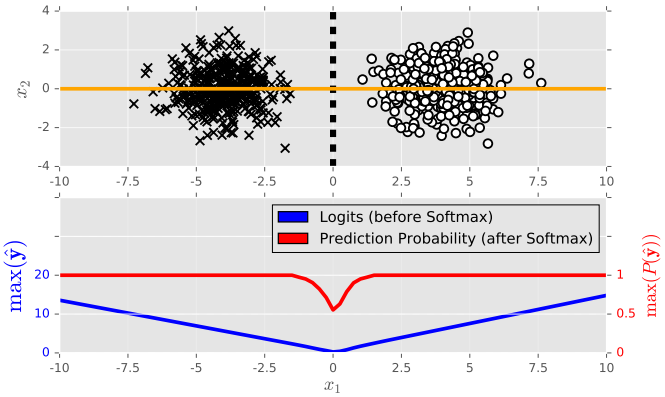


Fig. 4: A linearly separable two-class classification problem, highlighting the saturation of the softmax output, as opposed to the logit values, which keep increasing. Best viewed in color.

increases. However, the softmax output, after a certain point, remains the same and bounded.

Another demonstration of the aforementioned limitation of the softmax function for natural images can be observed in Fig. 2. In this figure, we provide an original image, classified as *arctic fox* with a confidence value of 0.99, and three adversarial counterparts, all of which are classified as *radio* with a confidence value of 1, and where classification is done by a pretrained ResNet-50 [1]. All three adversarial examples have been over-optimized in order to produce increasing logits at each step. However, the softmax output for all of them remains the same, masking the increase in the logit values.

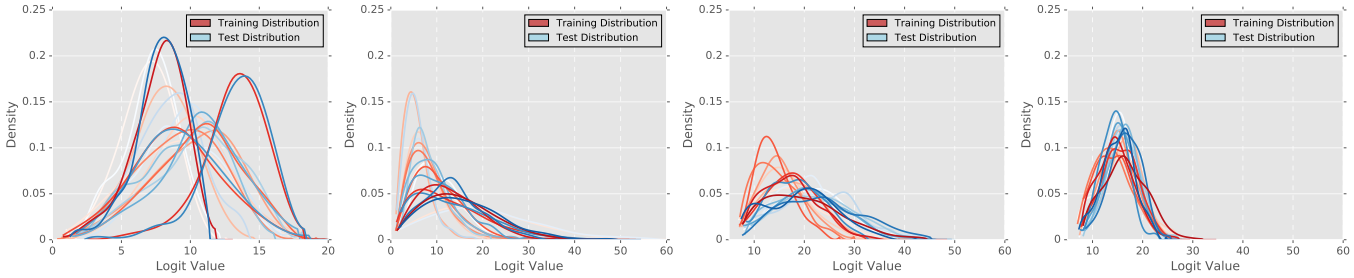
In Fig. 3, we present observations for multiple adversarial examples, showing the mean and the confidence interval of the highest logit value of 1000 adversarial examples and their corresponding softmax output. The 1000 adversarial examples were generated by making use of I-FGS and the L_2 version of the CW, using 100 optimization steps in a white-box setting. For I-FGS, we perturb the image one pixel at a time, and for CW, we use $K = 40$ in Eq. 1. The increase in the logit values for I-FGS is less pronounced than for CW since I-FGS relies on the signature of the gradient, which removes the precision in the perturbation between pixels. On the other hand, the spike in the confidence for I-FGS appears faster than for CW because the latter comes with a multi-class optimization nature and implements an extensive search process. Fig. 3 clearly shows the masking effect of softmax, given that the confidence almost immediately jumps to 100% after only a couple of iterations, making it from this point onwards impossible to differentiate between consecutive adversarial examples, whereas logit values keep increasing over the course of the optimization.

The results provided in Fig. 3 can be generalized to other targeted iterative adversarial example generation methods [5, 9, 4, 8, 16]. These iterative methods make it possible to further optimize an adversarial example (in terms of the logits), even after obtaining full confidence. However, once the prediction confidence is mapped to one, it is impossible to differentiate between the next iterations of the adversarial example based on the softmax output. Indeed, as the corresponding softmax input (i.e., logit values) keeps increasing, the softmax output will remain the same.

We refer the reader to our previous work which details the masking effect of the softmax function in the case of both single-target and multi-target adversarial attacks [21].

IV. Identifying Adversarial Examples

Targeted attacks, in every form, focus on maximizing an activation; for neural networks, this activation is the logit value for a particular class. Unless this optimization is stuck in a local maximum, we can say that $g(\theta, \mathbf{x}_{i+1})_c \geq g(\theta, \mathbf{x}_i)_c$, where c is the targeted class. If $\mathbf{x}$ is a data point that is the subject of box constraints, then this activation can only be maximized up to a certain point. Let $\mathbf{x}_r$ be a hypothetical genuine image that achieves the highest logit value for its category in supervised settings. We are interested in finding the numerical value of $g(\theta, \mathbf{x}_r)_c$, as this will allow us to label



(a) Dataset: MNIST
Model: LeNet-5

(b) Dataset: CIFAR-10
Model: Extended LeNet-5

(c) Dataset: ImageNet
Model: VGG-16

(d) Dataset: ImageNet
Model: ResNet-50

Fig. 5: Density plots of logit values of the prediction for ten classes, observed for both seen and unseen examples of the (a) MNIST, (b) CIFAR-10, and (c), (d) ImageNet datasets. We used LeNet-5 with ReLU activation [20], Extended LeNet-5 [12, 14], VGG-16 [22], and ResNet-50 [1] to obtain these results. For ImageNet, ten classes were selected randomly. Our models achieved 98%, 80%, 70.5%, and 77% top-1 accuracy on the respective datasets. These results are comparable to the results presented in [1, 12, 14, 22].

Dataset (Model)	1.5 IQR	2 IQR	3 IQR	4 IQR	5 IQR	$k_{\min}$
MNIST (LeNet-5)	0.1%	0.004%	**0%**	**0%**	**0%**	2.2
CIFAR-10 (Extended LeNet-5)	1.7%	0.6%	0.07%	0.01%	**0%**	4.9
ImageNet (VGG-16)	1.1%	0.3%	0.03%	0.004%	**0%**	4.6
ImageNet (ResNet-50)	0.7%	0.1%	0.009%	**0%**	**0%**	3.9
ImageNet (Inception-v3)	1.2%	0.4%	0.07%	0.001%	**0%**	4.9

TABLE I: Percentage of genuine images incorrectly identified as outliers, as obtained for different values of k in the following calculation: $g(\theta, \mathbf{x}) \overset{?}{>} Q_3 + k\ IQR$. Thresholds are calculated from correctly classified training examples for each class and tested on both training and test examples. k_{min} is the smallest value of k needed to ensure that none of the examples in the training and the test set are incorrectly identified as outliers.

any data point that produces a higher logit value than this value as an adversarial example without any further evaluation. To that end, in order to come up with an effective method to estimate $g(\theta, \mathbf{x}_r)_c$, we first analyze the prediction distributions of genuine images in terms of logit values.

A. Logit Distributions

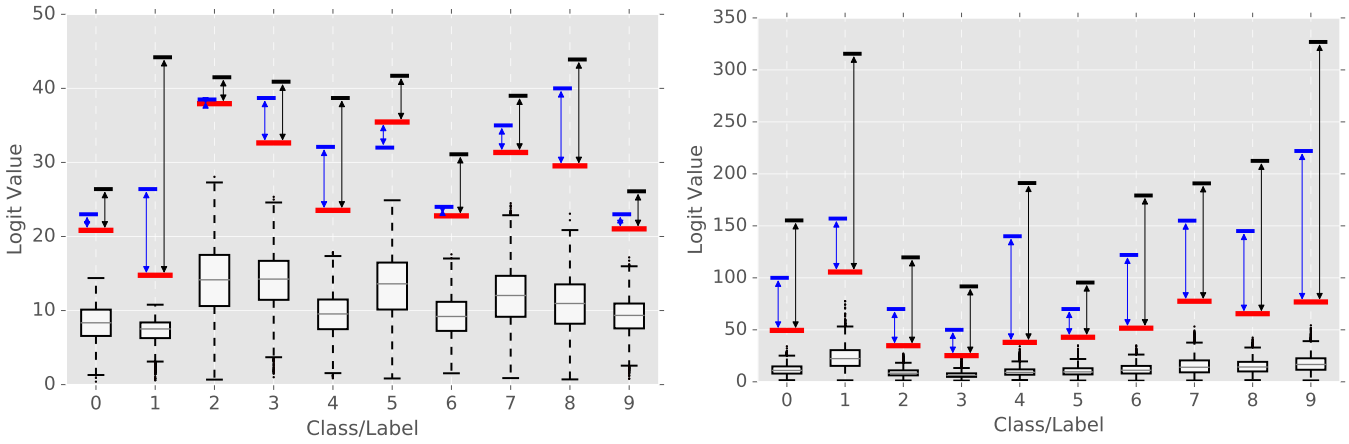
In order to come up with a robust method for countering adversarial examples (that is, a method that works for all models across all datasets), we first analyze the logit distribution of genuine images. A reoccurring problem in the field of adversarial examples is the usage of low-resolution images to show the effectiveness of a defense mechanism. Indeed, not all of the defense techniques for low-resolution images transfer to high-resolution images [14]. For this reason, we investigate the effectiveness of our method for MNIST, as well as for CIFAR-10 and ImageNet. In this context, we present Fig. 5, which shows the densities of predicted logit values for correctly classified samples of ten classes taken from MNIST, CIFAR-10, and ImageNet, for both training and test datasets. In the case of ImageNet, we present the distribution for both VGG-16 and ResNet-50. We can observe that the distributions of the logit values change between different datasets and between different classes/labels. On top of that, the distributions are vastly different for different architectures (VGG-16 and ResNet-50), even when the same dataset is used.

Keeping the aforementioned observations in mind, we arrive at the following conclusions: (a) the proposed method should be distribution-free so that it can be generalized across multiple datasets and multiple models, and (b) since the distribution of logit values also changes between different classes, the computational complexity of the proposed method should be low.

B. Determining the Threshold

Considering that datasets only contain a limited number of representations (per class), it is highly unlikely that the hypothetical genuine image that attains the highest activation is present in the dataset. However, we can estimate a threshold based on the activations of the observations at hand. A critical point in determining this threshold is to make sure that none of the existing images in the dataset is labeled as adversarial, as it is more important not to cast doubt on good observations than to miss an outlier [23, 24]. The same point is also highlighted by [14] when evaluating defense mechanisms.

As we showed previously, the logit value distributions are vastly different. Hence, when identifying outliers, we want to avoid any method that (implicitly or explicitly) makes assumptions on the data distribution. Therefore, based on the idea of boxplots, we propose using the interquartile range (IQR) to determine the threshold for identifying outliers (i.e., identifying adversarial examples). The IQR is defined as the difference between the 75th percentile (Q_3) and the 25th



(a) Dataset: MNIST
Model: LeNet-5

(b) Dataset: CIFAR-10
Model: Extended LeNet-5

Fig. 6: Illustration of adversarial subspaces with high logit values for (a) MNIST and (b) CIFAR-10. Logit value distributions of the predictions for the genuine images from the training set are given in the form of boxplots. The calculated thresholds and the highest logit value of the adversarial examples generated with I-FGS and the CW are highlighted as red, blue, and black lines, respectively. The adversarial spaces found using our method are indicated with arrows. Best viewed in color.

percentile (Q_1): $IQR = Q_3 - Q_1$. In basic statistical analysis, outliers are generally defined as those points that lie beyond the whiskers of the boxplot (i.e., below $Q_1 - k\ IQR$ or above $Q_3 + k\ IQR$). Traditionally, k is set to 1.5 [25], in which case the outliers are referred to as *mild*. We are, of course, only interested in large positive outliers. In this context, the authors of [23], [24], and [25] recommend using $k = 3$ for determining *extreme* outliers (i.e., highly unusual data points). Since we do not exactly know the underlying distribution of the logits, we experiment with different k values.

For each class, we calculate $Q_3 + k\ IQR$ as the threshold from the training sets of the aforementioned datasets and we present the percentage of misidentified images for the test datasets. Specifically, in Table I, we show the percentage of genuine images that are misindentified as *outliers* (i.e., adversarial examples) by the calculation $g(\theta, \mathbf{x}) \overset{?}{>} Q_3 + k\ IQR$, for different values of k. In the case of ImageNet, we present results for three different architectures, namely, VGG-16, ResNet-50, and Inception-V3 [22, 1, 26]. Based on these results, we can observe that $k = 3$ is mostly sufficient for most of the architectures, as the percentage of misidentified genuine images is, at most, as low as 0.07% (this corresponds to only 35 images in the test dataset of ImageNet). Nevertheless, it is also reasonable to prefer a threshold that does not reject *any* genuine image at all. In that regard, we can easily find $k_{\min}$ for different datasets and models, given that the proposed method is non-parametric in nature, which makes it straightforward to adopt this method for different problems. Additionally, by using the IQR instead of parameters such as the mean or standard deviation, which are sensitive to outliers and which make implicit assumptions about the underlying distributions, the proposed method is also more robust (i.e., not attracted by outliers). In the next section, we present experiments that show how the proposed method can be effectively used to detect over-optimized adversarial examples.

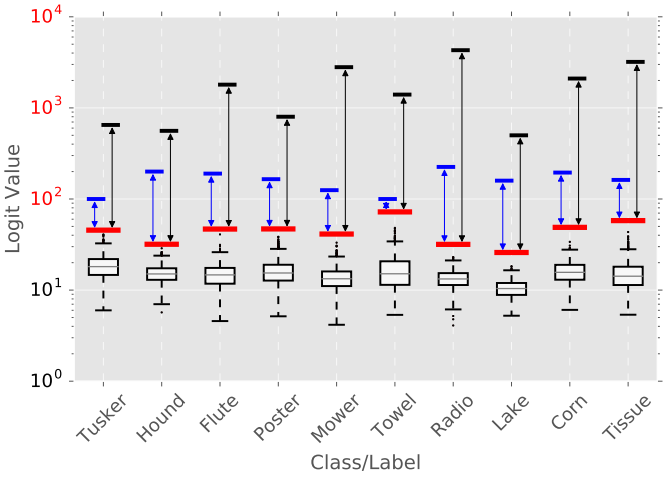


Fig. 7: Illustration of adversarial subspaces with high activation for the ImageNet dataset using ResNet-50. Logit value distributions of the predictions for the genuine images from the training set are given in the form of boxplots. The calculated thresholds and the highest logit value of the adversarial examples generated with I-FGS and the CW are highlighted as red, blue, and black lines, respectively. The adversarial spaces found using our method are indicated with arrows. Note that the y-axis is log-scaled for clarity. Best viewed in color.

V. Experimental Results

Using $T = Q_3 + k\ IQR$, we calculated logit thresholds that do not leave out any genuine images for MNIST, CIFAR-10, and ImageNet, using the $k_{\min}$ values given in Table I. For each of the selected classes in those datasets (all classes in the case

of MNIST and CIFAR-10; ten randomly selected classes in the case of ImageNet), we generate 500 adversarial examples using I-FGS and the L_2 version of the CW, totalling up to 5000 adversarial examples for each dataset, with the aim of finding an adversarial example that generates the highest logit value for that class. By doing so, we want to determine the space for which we can identify *all* adversarial examples that lie between the proposed threshold and the produced logit limit for the adversarial example generation methods.

Fig. 6 and Fig. 7 show the results obtained for MNIST, CIFAR-10, and ImageNet (with ResNet-50), with the predicted logit values given in the form of boxplots and with the calculated thresholds highlighted with red lines. The adversarial examples that generate the highest logit prediction values are highlighted separately for the two adversarial example generation methods, using a blue line for I-FGS and a black line for CW. We annotate the space between the proposed threshold (red) and the maximum logit value for each adversarial example generation method $g_c(\theta, \mathbf{x}) > T$ to show that we can immediately identify numerous adversarial examples within this space, no matter which method is used to generate these adversarial examples. As expected, for all three datasets, the CW generates *stronger* adversarial examples that produce higher logit values than I-FGS.

Note that, as the image resolution increases, the upper limit for the logit values that can be produced by an adversarial example also increases. From MNIST to CIFAR-10, this limit increases by a factor of 7, and from MNIST to ImageNet, this limit increases by a factor of 200. In the case of the ImageNet dataset, the increase is so high that we present the y-axis of the graph in the base ten logscale for the sake of readability.

A. Feasibility for Higher Resolution Images

As shown by Fig. 6 and Fig. 7, when the resolution of an image increases, the space in which adversarial examples can be generated also increases in a significant way. This property is also highlighted by other studies [27].

Let us define, for a given model, the effectiveness of our method as

$$D_{Adv} = \frac{1}{M} \sum_{c=1}^{M} \frac{g(\theta, (\mathbf{x}_{Adv})_c)_c - T_c}{g(\theta, (\mathbf{x}_{Adv})_c)_c}, \tag{2}$$

where M is the number of classes in the dataset, c is the current target class, $(\mathbf{x}_{Adv})_c$ is the adversarial example that produces the highest logit value when targeting class c, and $T_c = (Q_3 + k_{min}\ IQR)_c$ is the calculated threshold for the targeted class. For each class, we calculate the space between the threshold T_c and the logit value of $(\mathbf{x}_{Adv})_c$ and normalize it by dividing by the length of the total space. This value corresponds to the proportion of the adversarial space for class c with respect to the size of the total target class that we can detect by our method. We then take the average over all classes to get an approximate idea of the proportion of the adversarial space we can detect in function of the total dataset. Thanks to the normalization factor, D_{Adv} will allow us to compare different dataset/model combinations, where a higher number indicates a higher effectiveness of detecting adversarial examples.

Dataset (Model)	Image Resolution	k_{min}	Space Covered $D_{Adv} \in [0, 1]$
MNIST (LeNet-5)	$1 \times 28 \times 28$	2.2	26%
CIFAR-10 (Ext. LeNet-5)	$3 \times 32 \times 32$	4.9	68%
ImageNet (VGG-16)	$3 \times 224 \times 224$	4.6	75%
ImageNet (ResNet-50)	$3 \times 224 \times 224$	3.9	79%
ImageNet (Inception-v3)	$3 \times 299 \times 299$	4.9	89%

TABLE II: Approximate proportion of the number of adversarial examples detected in each dataset (calculated using Eq. 2).

Applying this formula allows us to construct Table II, which shows the proportion of potential adversarial examples detected. Based on the results presented in Table II, we observe that the proposed method is able to detect more adversarial examples when the resolution of an image is higher.

B. Scalability and Computational Cost

In Section IV-B, we noted that the thresholds must be calculated individually for each class. Naturally, when the number of classes increases, the number of thresholds that must be calculated also increases. However, since these thresholds only need to be calculated and stored once, the computational cost of the proposed method is small; it only requires one full forward pass over the train dataset, after which the thresholds can be used indefinitely.

Another strength of the proposed method is the easiness with which thresholds can be updated. When the size of a training set increases (thanks to an influx of additional data), the thresholds can simply be re-calculated.

C. Defense Against Adversarial Examples

In Section V and Section V-A, we put the emphasis on the number of potential adversarial examples countered using the proposed method. However, only using the proposed method is not a viable option to identify all adversarial examples. This is especially true in white-box settings, for which it is easy to come up with an attack that would bypass our method. Indeed, one can simply introduce a logit constraint when generating adversarial examples, making sure the generated adversarial examples produce lower logit values than the proposed thresholds. Nonetheless, our intention in proposing this method is not to use it as a universal method for detecting all adversarial examples, but rather to add it to a particular defense workflow as a first (and computationally inexpensive) line of defense for identifying adversarial examples, with these adversarial examples producing unnatural logit values that are beyond the reach of any genuine images.

Applying the proposed method ensures that adversarial example generation methods will be limited not only by the amount of perturbation and the discretization constraint, but also by the maximum logit value produced by an adversarial example. This will limit the search space extensively, especially when the resolution of the images under consideration is large, as is shown in Section V-A.

VI. Conclusion and Future Research

In this paper, we established the fundamentals for the use of logit values as an indicator of adversariality. To that end, we first discussed the masking effect of the softmax function, showing that the logit values keep increasing, even after softmax output achieving its maximum value of one during the generation of adversarial examples.

Next, we laid out examples of logit distributions from multiple datasets and showed the need for a distribution-free method to identify adversarial examples, leading to the introduction of a non-parametric and computationally cheap technique for detecting over-optimized adversarial examples. Throughout this paper, we presented experimental results for MNIST, CIFAR-10, and ImageNet, also using different neural network architectures.

The main purpose of the newly introduced defense technique is to further limit the expressiveness of methods for adversarial example generation, not only in terms of perturbation amount, but also in terms of logits. This allows our technique to be used as a first line of defense, before triggering a well-rounded one that is more complex in nature. Therefore, future research may focus on investigating the compatibility of our technique with other defense mechanisms that are complimentary in nature.

Acknowledgements

The research activities described in this paper were funded by Ghent University Global Campus, Ghent University, imec, Flanders Innovation & Entrepreneurship (VLAIO), the Fund for Scientific Research-Flanders (FWO-Flanders), and the EU.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[2] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *CoRR*, vol. abs/1506.02640, 2015.

[3] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Springer International Publishing, 2015, pp. 234–241.

[4] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *CoRR*, vol. abs/1312.6199, 2013.

[5] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 427–436.

[6] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: A simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2574–2582.

[7] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *CoRR*, vol. abs/1412.6572, 2014.

[8] N. Papernot, P. D. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," *CoRR*, vol. abs/1511.07528, 2015.

[9] N. Carlini and D. A. Wagner, "Towards evaluating the robustness of neural networks," *CoRR*, vol. abs/1608.04644, 2016.

[10] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *CoRR*, vol. abs/1611.01236, 2016.

[11] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[12] N. Papernot, P. D. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," *CoRR*, vol. abs/1511.04508, 2015.

[13] J. Lu, T. Issaranon, and D. A. Forsyth, "Safetynet: Detecting and rejecting adversarial examples robustly," *CoRR*, vol. abs/1704.00103, 2017.

[14] N. Carlini and D. A. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," *CoRR*, vol. abs/1705.07263, 2017.

[15] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," *arXiv preprint arXiv:1802.00420*, 2018.

[16] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *CoRR*, vol. abs/1607.02533, 2016.

[17] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[18] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[19] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in *Neurocomputing*. Springer, 1990, pp. 227–236.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[21] U. Ozbulak, W. De Neve, and A. Van Messem, "How the softmax output is misleading for evaluating the strength of adversarial examples," *NeuRIPS 2018, Workshop on Security in Machine Learning, arXiv:1811.08577*, 2018.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[23] M. Frigge, D. C. Hoaglin, and B. Iglewicz, "Some im-

plementations of the boxplot," *The American Statistician*, vol. 43, no. 1, pp. 50–54, 1989.

[24] F. R. Hampel, "The breakdown points of the mean combined with some rejection rules," *Technometrics*, vol. 27, no. 2, pp. 95–107, 1985.

[25] J. W. Tukey, *Exploratory Data Analysis*. Reading, Mass., 1977, vol. 2.

[26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[27] C.-J. Simon-Gabriel, Y. Ollivier, B. Schölkopf, L. Bottou, and D. Lopez-Paz, "Adversarial vulnerability of neural networks increases with input dimension," *arXiv preprint arXiv:1802.01421*, 2018.